What is claimed is:

1. A method for monitoring the performance of applications running on a plurality of servers in a distributed computing environment, comprising the steps of: receiving from a user selected information for monitoring, monitoring application performance in accordance with the selected information, and making monitored performance information available to the user in accordance with the selected information.
2. The method of claim 1, wherein the user is prompted to identify a scope of information to be monitored, and wherein application server performance is monitored in accordance with the selected scope.
3. The method of claim 2, wherein said scope comprises a first monitoring level wherein the selected information comprises request level data and server level data.
4. The method of claim 3, wherein said scope further comprises a second monitoring level wherein the selected information further comprises API level data.
5. The method of claim 4, wherein said scope further comprises a third monitoring level, wherein the selected information further comprises method level data.
6. The method of claim 1, wherein the user is prompted to identify a schedule for monitoring of information, and wherein application server performance is monitored in accordance with the identified schedule.

7. A method for monitoring the performance of applications running on a plurality of servers in a distributed computing environment, comprising the steps of prompting a user to select a server or server group, a resource, and a threshold or condition for notification, comparing the value or quality of a parameter to the threshold or condition, and, if the parameter reaches the threshold or condition, logging information concerning the parameter.
8. The method of claim 7, further comprising the step of comparing the parameter to criteria for notifying a user, and notifying a user if the criteria are met.
9. The method of claim 7, wherein the threshold or condition is a value of a resource.
10. The method of claim 9, wherein the resource is a property of a method.
11. The method of claim 10, wherein the resource is CPU time.
12. The method of claim 7, wherein the threshold or condition is a number of hits.

13. The method of claim 12, wherein the resource is a request, and the condition is a string contained in the request.
14. The method of claim 7, wherein the resource relates to application server performance, and the condition is percentage of CPU time.
15. A method for monitoring the performance of applications running on a plurality of servers in a distributed computer system, comprising the steps of providing the user with performance information, receiving from the user a request for more specific performance information, and providing more detailed performance information.
16. The method of claim 15, further comprising the step of providing the user a selection of request analysis, method analysis, SQL analysis, server availability analysis, and application server analysis, and receiving a selection from the user.
17. The method of claim 16, further comprising the step of providing a trend analysis in accordance with the received selection.
18. The method of claim 17, further comprising the step of receiving a request for a decomposition report on a portion of the trend analysis, and providing a decomposition report.
19. The method of claim 18, further comprising the steps of receiving a request for a detail report on portion of the decomposition report, and providing a detail report.
20. A method for monitoring the performance of applications running on a plurality of servers in a distributed computer system, comprising the steps of receiving from the user an identification of a server as an authoritative server, another server as a comparison server, comparing runtime environment data and displaying a comparison.
21. The method of claim 20, wherein said runtime environment data is in the form of one or more of CPU data, server data and Java data between the selected servers.
22. The method of claim 21, wherein said CPU data comprises one or more of CPU speed, number of CPUs online, number of CPUs offline, memory, operating system version, and physical disk space.
23. The method of claim 21, wherein said server data comprises one or more of application server, startup directory, listening ports, SSL listening ports, number of

registered servlets, number of registered EJBs, number of JDBC connection pools, and number of EARs.

24. The method of claim 21, wherein said Java data comprises one or more of JDK version, installation directory, Java policy, operating system information, class path, and library path.
25. A method for monitoring the performance of applications running on a plurality of servers in a distributed computer system, comprising the steps of receiving from the user an identification of a server as an authoritative server, and another server as a comparison server, and providing a list of matching and differing file names.
26. The method of claim 25, further comprising the step of comparing file size.
27. The method of claim 25, further comprising the step of comparing file time stamp.
28. The method of claim 25, further comprising the steps of receiving from the user a selection of a file, conducting a comparison of the files, and providing a result to the user.
29. The method of claim 28, wherein the step of conducting a comparison of the files further comprises conducting a comparison of the files via MD5 checksum calculation.
30. A method for monitoring the performance of applications running on a plurality of servers in a distributed computer system, comprising the steps of assigning a role to each user, and mapping between access to functions and each user role by an access control list, whereby access to functions is limited depending on the assigned role of the user.
31. The method of claim 30, wherein one of said assigned roles is administrator, said administrator having read/write access to each of the functions.
32. The method of claim 30, wherein one of said assigned roles is user, said user having display access only to each of the functions.
33. A method for processing of requests, comprising the steps of receiving a request string, and mapping the received request string to a distinguishable request string and a collapsible request string.
34. The method of claim 33, wherein the received request string is in the form of one of a JSP, a servlet, and remote Enterprise Java Bean calls.

35. The method of claim 33, further comprising the steps of prompting a user to create rules for mapping of a received request string to a distinguishable request string and a collapsible request string, receiving rules in response to the step of prompting, and applying the received rules.

36. A method for monitoring of performance of applications in a distributed environment, comprising the steps of providing, in a system running at least one application, a management application having various components for monitoring and management, and monitoring and providing to a user in real-time information concerning configuration of the components and the relationships between the components.

37. A system for monitoring the performance of applications running on a plurality of servers in a distributed computing environment, comprising means for monitoring application performance in accordance with the selected information received from a user, and means for making monitored performance information available to the user in accordance with the selected information.

38. The system of claim 37, further comprising means for prompting a user to identify a scope of information to be monitored, and means for monitoring application server performance in accordance with the selected scope.

39. The system of claim 38, wherein said scope comprises a first monitoring level wherein the selected information comprises request level data and server level data.

40. The system of claim 39, wherein said scope further comprises a second monitoring level wherein the selected information further comprises API level data.

41. The system of claim 40, wherein said scope further comprises a third monitoring level, wherein the selected information further comprises method level data.

42. The system of claim 37, further comprising means for monitoring application server performance in accordance with an identified schedule received from a user.

43. A system for monitoring the performance of applications running on a plurality of servers in a distributed computing environment, comprising means for prompting a user to select a server or server group, a resource, and a threshold or condition for notification, means for comparing a value or quality of a parameter on the selected server or server group to the selected threshold or condition, and, means for logging

information concerning the selected parameter if the selected parameter reaches the selected threshold or condition.

44. The system of claim 43, further comprising means for comparing the parameter to criteria for notifying a user, and means for notifying a user if the criteria are met.
45. The system of claim 43, wherein the threshold or condition is a value of a resource.
46. The system of claim 45, wherein the resource is a property of a method.
47. The system of claim 46, wherein the resource is CPU time.
48. The system of claim 43, wherein the threshold or condition is a number of hits.
49. The system of claim 48, wherein the resource is a request, and the condition is a string contained in the request.
50. The system of claim 43, wherein the resource relates to application server performance, and the condition is percentage of CPU time.
51. A system for monitoring the performance of applications running on a plurality of servers in a distributed computer system, comprising means for providing the user with performance information, and means for providing more detailed performance information in response to a user request for more detailed performance information.
52. The system of claim 51, further comprising means for providing the user a selection of request analysis, method analysis, SQL analysis, server availability analysis, and application server analysis, and means for receiving a selection from the user.
53. The system of claim 52, further comprising means for providing a trend analysis in accordance with the received selection.
54. The system of claim 53, further comprising means for receiving a request for a decomposition report on a portion of the trend analysis, and means for providing a decomposition report.
55. The system of claim 53, further comprising means for receiving a request for a detail report on portion of the decomposition report and means for providing a detail report.
56. A system for monitoring the performance of applications running on a plurality of servers in a distributed computer system, comprising means for receiving from the user an identification of a server as an authoritative server, another server as a comparison server, means for comparing runtime environment data and means for displaying a comparison.

57. The system of claim 56, wherein said runtime environment data is in the form of one or more of CPU data, server data and Java data between the selected servers.

58. The system of claim 57, wherein said CPU data comprises one or more of CPU speed, number of CPUs online, number of CPUs offline, memory, operating system version, and physical disk space.

59. The system of claim 57, wherein said server data comprises one or more of application server, startup directory, listening ports, SSL listening ports, number of registered servlets, number of registered EJBs, number of JDBC connection pools, and number of EARs.

60. The system of claim 57, wherein said Java data comprises one or more of JDK version, installation directory, Java policy, operating system information, class path, and library path.

61. A system for monitoring the performance of applications running on a plurality of servers in a distributed computer system, comprising means for, in response to a received identification of a first server as an authoritative server and second server as a comparison server, providing a list of matching and differing file names in said authoritative and comparison servers.

62. The system of claim 61, further comprising means for comparing file size.

63. The system of claim 61, further comprising means for file time stamps.

64. The system of claim 61, further comprising means for conducting a comparison of files selected by a user, and providing a result to the user.

65. The system of claim 64, wherein said means for conducting a comparison comprises means for conducting a comparison of the files via MD5 checksum calculation.

66. A system for monitoring the performance of applications running on a plurality of servers in a distributed computer system comprises means for assigning a role to each user, and means for mapping between access to functions and each user role by an access control list, whereby access to functions is limited depending on the assigned role of the user.

67. The system of claim 66, wherein one of said assigned roles is administrator, said administrator having read/write access to each of the functions.

68. The system of claim 66, wherein one of said assigned roles is user, said user having display access only to each of the functions.

69. A system for handling requests, comprising means for receiving a request string, and means for mapping the received request string to a distinguishable request string and a collapsible request string.

70. The system of claim 69, wherein the received request string is in the form of one of a JSP, a servlet, and a remote Enterprise Java Bean call.

71. The system claim 69, further comprising means for prompting a user to create rules for mapping of a received request string to a distinguishable request string and a collapsible request string, and means for applying rules received in response to a prompt to a user to create rules for mapping of a received request string to a distinguishable request string and a collapsible request string.

72. A system for monitoring of performance of applications in a distributed environment, comprising means for providing, in a system running at least one application, a management application having various components for monitoring and management, and means for monitoring and providing to a user in real-time information concerning configuration of the components and the relationships between the components.

73. A computer program for monitoring the performance of applications running on a plurality of servers in a distributed computing environment, said program consisting of instructions stored on a medium, said instructions, when executed on a processor causing the processor to execute the steps of: receiving from a user selected information for monitoring, monitoring application performance in accordance with the selected information, and making monitored performance information available to the user in accordance with the selected information.

74. A computer program for monitoring the performance of applications running on a plurality of servers in a distributed computing environment, said program consisting of instructions stored on a medium, said instructions, when executed on a processor causing the processor to execute the steps of: prompting a user to select a server or server group, a resource, and a threshold or condition for notification, comparing the value or quality of a parameter to the threshold or condition, and, if the parameter reaches the threshold or condition, logging information concerning the parameter.

75. A computer program for monitoring the performance of applications running on a plurality of servers in a distributed computer system, said program consisting of instructions stored on a medium, said instructions, when executed on a processor causing the processor to execute the steps of: providing the user with performance information, receiving from the user a request for more specific performance information, and providing more detailed performance information.

76. A computer program for monitoring the performance of applications running on a plurality of servers in a distributed computer system, said program consisting of instructions stored on a medium, said instructions, when executed on a processor causing the processor to execute the steps of: receiving from the user an identification of a first server as an authoritative server, of a second server as a comparison server, comparing runtime environment data from said authoritative and comparison servers and displaying a comparison.

77. A computer program for monitoring the performance of applications running on a plurality of servers in a distributed computer system, said program consisting of instructions stored on a medium, said instructions, when executed on a processor causing the processor to execute the steps of: receiving from the user an identification of a first server as an authoritative server, and a second server as a comparison server, and providing a list of matching and differing file names on said authoritative and comparison servers.

78. A computer program for monitoring the performance of applications running on a plurality of servers in a distributed computer system, said program consisting of instructions stored on a medium, said instructions, when executed on a processor causing the processor to execute the steps of: assigning a role to each user, and mapping between access to functions and each user role by an access control list, whereby access to functions is limited depending on the assigned role of the user.

79. A computer program for processing of requests, said program consisting of instructions stored on a medium, said instructions, when executed on a processor causing the processor to execute the steps of: receiving a request string, and mapping the received request string to a distinguishable request string and a collapsible request string.

80. A computer program for monitoring of performance of applications in a distributed environment, comprising the steps of providing, in a system running at least one application, a management application having various components for monitoring and management, and monitoring and providing to a user in real-time information concerning configuration of the components and the relationships between the components.